

Partitioning has become vitally important in managing information technology resources. PSI's system employs partitioning to allocate its resources in ways that infringe IBM's patents.

A. The '812 "Partition Communication" Patent

Computers in a network can communicate by sending messages or "packets" to each other over a network. Before the invention of IBM's '812 patent, if two partitions of the same computer sought to communicate with each other over a network, the message would be routed out onto the network and then back to the originating computer. The '812 patent allows these partitions to communicate internally without the time-consuming step of sending messages over the network. To do so, the network interface saves each partition's address for later use. Then, when one partition attempts to communicate with another, the system is aware of the destination partition's address and the partitions can communicate internally without using the network.

The parties now dispute the construction of the claim terms "host network interface" and "saving at said host-network interface." The parties also dispute the identification of structure in the specification of the '812 patent that corresponds to certain limitations of claim 11 that are in means-plus-function format.

1. "host-network interface" (claims 1, 11)

IBM's CONSTRUCTION	PSI's CONSTRUCTION
The communication interface between a host computer and a network.	A hardware component coupled between a mainframe class data processing system and a network port.

The term "host-network interface" was well-known to a person of ordinary skill in the art at the time of the invention, and has a plain and ordinary meaning reflected in IBM's proposed construction. The plain and ordinary meaning of "host network interface" is an interface between a host computer and a network. This plain and ordinary meaning is confirmed by several dictionary definitions:

host interface = the interface between a communications network and a host computer.

(Ex. 43, *The Authoritative Dictionary of IEEE Standards Terms*, 7th Ed., p. 523 (2000).)

host interface = interface between a network and a host computer.

(Ex. 44, *IBM Dictionary of Computing*, 10th Ed., p. 318 (1994).) Given this easily understood meaning, the '812 claims, specification, and prosecution history do not define the term, but rather describe an embodiment including a host-network interface.

The parties' proposed constructions are very similar, with the three differences highlighted in the table below:

IBM's CONSTRUCTION	PSI's CONSTRUCTION
The <i>communication interface</i> between	A <i>hardware component</i> coupled between
... <i>a host computer</i> <i>a mainframe class data processing system</i> ...
...and a network.	...and a network <i>port</i> .

Each part of IBM's proposed construction is more consistent with the intrinsic and extrinsic evidence and the plain and ordinary meaning of the term.

(a) "communication interface" v. "hardware component"

IBM's construction ("communication interface") is consistent with the plain and ordinary meaning of "host-network interface," as well as the '812 claims, specification, and prosecution history. While PSI limits the host-network interface to hardware, under IBM's construction the host-network interface can be implemented in both hardware and software.

Claim 1 of the '812 patent introduces a "host-network interface" in the preamble:

A method of network communications implemented within a host-network interface for use in a mainframe class data processing system having multiple partitions and a port to a network, said method comprising:

('812 patent, col. 11:6-9.) Program product claim 11 also recites that all three steps of the claimed invention (*i.e.*, the method of claim 1) are performed by "computer readable program code." Thus, it is clear from the language of the claims that "host-network interface" can be implemented in software and, therefore, is not limited to being a "hardware component" as PSI now asserts.

This is further confirmed by the '812 specification, which describes (and illustrates in Fig, 3) one embodiment of a host-network interface as a "host channel connection 118 and a communications adapter, such as an IBM Open Systems Adapter (OSA) 120." ('812 patent, col. 5:12-13.)

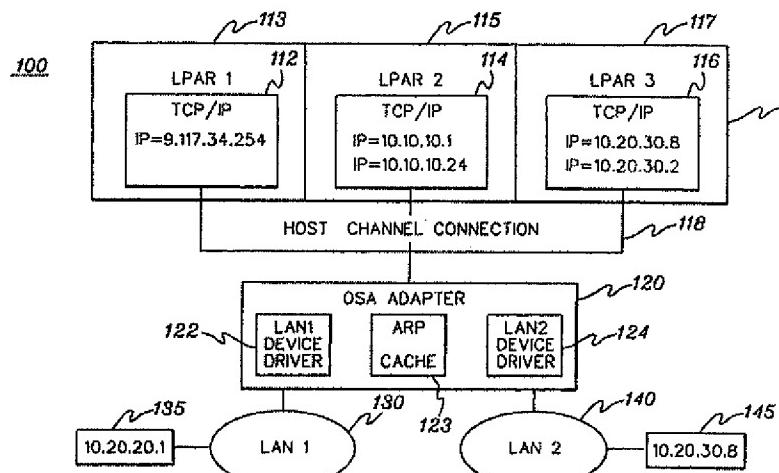


fig. 3

This embodiment consists of both hardware and software, because the Open Systems Adapter (OSA) 120 includes LAN Device Drivers 122 and 124, which are software components. (See '812 patent, col. 5:52-6:13; Ex. 45, *IBM Dictionary of Computing*, 10th Ed., p. 193 (1994); Ex. 46, *The Authoritative Dictionary of IEEE Standard Terms*, 7th Ed., p. 298 (2000); Smotherman Decl. ¶¶ 19-21; see also *SanDisk Corp. v. Memorex Products, Inc.*, 415 F.3d 1278, 1285 (Fed. Cir. 2005) ("A claim construction that excludes a preferred embodiment, moreover, 'is rarely, if ever, correct.'") (*citation omitted*).

Because the intrinsic evidence expressly includes software as part of the host-network interface, PSI's proposed construction for interface is too narrow, and the Court should adopt IBM's construction.

(b) "host computer and a network" v. "mainframe class data processing system and a network port"

As discussed above, the plain and ordinary meaning of a host-network interface is an interface between a host (or host computer) and a network. IBM's proposed construction

precisely follows this plain and ordinary meaning and is consistent with the intrinsic and extrinsic evidence. PSI's proposed construction, on the other hand, improperly attempts to limit the construction of "host" and "network" to "mainframe class data processing system" and "network port," without intrinsic or extrinsic support.

Returning to the preamble of claim 1, this part of the claim recites "A method of network communications implemented within a host-network interface *for use in a mainframe class data processing system* having multiple partitions and a *port* to a network, said method comprising: . ." ('812 patent, col. 11:6-9.) Thus, the language "mainframe class data processing system" and "port" that PSI is attempting to read into the term "host network interface" already appear in claim 1 separately and as a modifier to the term "host network interface." (*Id.*, col. 11:7-9.) PSI's construction should be rejected because if "host network interface" were limited by the language PSI's proposes, then there would be no reason for the claim drafter to have included that language separately in claim 1. PSI is attempting to insert non-limiting preamble language into a narrow construction of "host-network interface," even though it is already set forth in the claim and would be superfluous. *See, e.g., Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006) ("claims are interpreted with an eye toward giving effect to all terms in the claim."); *NTP, Inc. v. Research in Motion, Ltd.*, 418 F.3d 1282, 1300 (Fed. Cir. 2005) (rejecting patentee's attempt to interject term into a construction when that term already appeared in the claim.).

The '812 specification and prosecution history do not limit either the "host" to a "mainframe class data processing system" or the "network" to a "network port," and while the preferred embodiments disclose a "mainframe environment," these embodiments are not limiting. (*See* '812 patent, col. 10:66-11:04.)

IBM's proposed construction is consistent with the plain and ordinary meaning of the term "host-network interface," as well as the intrinsic and extrinsic evidence, and should be adopted.

2. "saving at said host-network interface" (claim 1)

IBM's CONSTRUCTION	PSI's CONSTRUCTION
Saving at said host-network interface.	Saving on memory located within the host-network interface.

The first element of claim 1 includes the phrase "saving at said host-network interface": saving at said host-network interface an internet protocol (IP) address of at least one of said multiple partitions of the mainframe class data processing system; ('812 patent, col. 11:10-12.) IBM believes this phrase should be given its plain and ordinary meaning. The term "host-network interface" has already been construed, leaving just the term "saving," which the parties agree should be given its plain and ordinary meaning.

PSI's construction improperly attempts to limit saving the IP address "on memory located within the host-network interface." This is contrary to the language of claim 1, which simply does not require that the IP address be saved "on memory located within" the host-network interface. Instead, PSI is improperly attempting to import limiting language from the '812 specification. *See Phillips*, 415 F.3d. at 1323 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.").

Accordingly, this phrase needs no construction other than its plain and ordinary meaning.

3. "at least one computer usable medium having computer readable program code means embodied therein for causing network communications in a mainframe class data processing system having multiple partitions and a port to a network, the computer readable program code means in the article of manufacture comprising" (claim 11)

The parties agree that this element is governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Causing network communications in a mainframe class data processing system having multiple partitions and a port to a network.</p> <p>Structure: An article of manufacture (<i>see, e.g.,</i> 10:46-57 and equivalents thereof) that causes a computer to cause network communications in a mainframe class data processing system having multiple partitions and a port to a network (<i>see, e.g.,</i> Figs. 5, 6, 7, 8A, 8B; or col. 7:45-8:19; 8:31-48; 9:8-10; 9:46-57; 10:1-11), and equivalents thereof.</p>	<p>Function: Causing network communications in a mainframe class data processing system having multiple partitions and a port to a network.</p> <p>Structure: No corresponding structure.</p>

Structure. Claim 11 reads as follows:

11. An article of manufacture comprising:
at least one computer usable medium having computer readable program code means embodied therein for causing network communications in a mainframe class data processing system having multiple partitions and a port to a network, the computer readable program means in the article of manufacture comprising:
- (i) computer readable program code means for causing a computer to effect saving at a host-network interface an internet protocol (IP) address of at least one of the multiple partitions of the mainframe class data processing system;
 - (ii) computer readable program code means for causing a computer to effect generating an IP datagram at a first partition of said multiple partitions to be forwarded to a second partition of said multiple partitions using a destination IP address; and
 - (iii) computer readable program means for causing a computer to effect determining whether said destination IP address for said IP datagram comprises an IP address saved at said host-network interface for said at least one partition, and if so, forwarding the IP datagram directly from said first partition to said second partition of said multiple partitions without employing said network.

The first element of this claim (beginning with the phrase "at least one computer usable medium") identifies the "computer readable program code means," which are then set forth in the second, third, and fourth elements of the claim. As a result, the structure analysis for this

element must consider the structure set forth in the three referenced elements. Accordingly, IBM will address in Sections (i), (ii), and (iii) below the structure in those three referenced elements.

- (i) "computer readable program code means for causing a computer to effect saving at a host-network interface an internet protocol (IP) address of at least one of the multiple partitions of the mainframe class data processing system" (claim 11)**

The parties agree this is a means-plus-function element governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Causing a computer to effect saving at a host-network interface an internet protocol (IP) address of at least one of the multiple partitions of the mainframe class data processing system.</p> <p>Structure: An article of manufacture (<i>see, e.g.</i>, 10:46-57) that causes a computer to effect saving at a host-network interface an internet protocol (IP) address of at least one of the multiple partitions of the mainframe class data processing system (<i>see, e.g.</i>, Fig. 5, 7:66-8:4), and equivalents thereof.</p>	<p>Function: Causing a computer to effect saving at a host-network interface an internet protocol (IP) address of at least one of the multiple partitions of the mainframe class data processing system.</p> <p>Structure: No corresponding structure.</p>

Structure. The agreed-upon function is performed by the computer readable program code stored on a computer usable medium. The '812 specification discloses the structure for this computer usable medium:

Additionally, at least one program storage device readable by machine, tangibly embodying at least one program of instructions executable by the machine, to perform the capabilities of the present invention, can be provided.

('812 patent, col. 10:53-57.)

The '812 specification also discloses an algorithm for this computer readable program code to cause a computer to perform the recited function, as follows:

At initialization time, each Host TCP/IP stack is configured to register its HOME IP addresses with the OSA adapter in a manner apparent to one skilled in the art.

The "HOME" addresses are those which are recognized as local IP addresses by the specific stack. These entities are marked as HOME entries in the ARP cache.

('812 patent, col. 7:66-8:4.) This "saving" of an IP address as a HOME IP address is illustrated by Fig. 5:

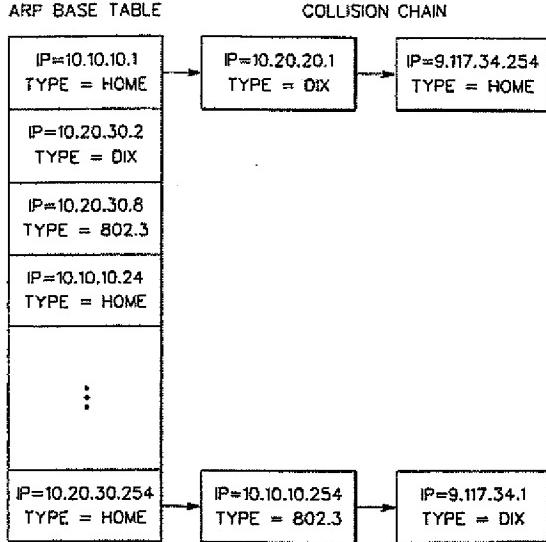


fig. 5

Thus, the '812 specification discloses structure and the corresponding algorithm for causing a computer to perform the recited function, and the Court should adopt this construction.

PSI asserts that the '812 specification provides "no corresponding structure." This is contrary to the plain language of the '812 specification, which discloses the structure and the corresponding algorithm for causing a computer to perform the recited function. Thus, the Court should reject PSI's construction.

- (ii) "**computer readable program code means for causing a computer to effect generating an IP datagram at a first partition of said multiple partitions to be forwarded to a second partition of said multiple partitions using a destination IP address**" (claim 11)

The parties agree that this phrase is governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

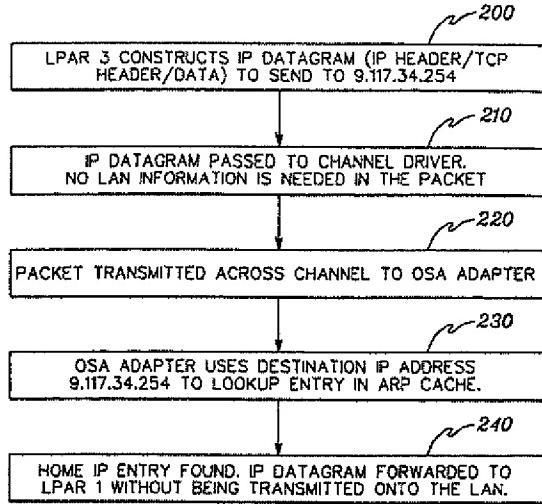
IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Causing a computer to effect generating an IP datagram at a first partition of said multiple partitions to be forwarded to a second partition of said multiple partitions using a destination IP address.</p> <p>Structure: An article of manufacture (<i>see, e.g.,</i> 10:46-57) that causes a computer to effect generating an IP datagram at a first partition of said multiple partitions to be forwarded to a second partition of said multiple partitions using a destination IP address (<i>see, e.g.,</i> step 200 of Fig. 6, 8:33-37; step 300 of Fig. 7, 9:8-10; Fig. 8A, 9:46-57; or Fig. 8B, 10:1-11), and equivalents thereof.</p>	<p>Function: Causing a computer to effect generating an IP datagram at a first partition of said multiple partitions to be forwarded to a second partition of said multiple partitions using a destination IP address.</p> <p>Structure: No corresponding structure.</p>

Structure. The agreed-upon function is performed by the computer readable program code stored on a computer usable medium. The '812 specification discloses the structure for this computer usable medium:

Additionally, at least one program storage device readable by machine, tangibly embodying at least one program of instructions executable by the machine, to perform the capabilities of the present invention, can be provided.

('812 patent, col. 10:53-57.)

The '812 specification also discloses an algorithm for this computer readable program code to cause a computer to perform the recited function in several places. For example, step 200 of Fig. 6 discloses "LPAR 3" (e.g., "a first partition") generating an IP datagram:

*fig. 6*

(See also, '812 patent, col. 8:33-37.) The '812 specification also describes this generation of an IP datagram in Fig. 7, step 300 ("LPAR 3 CONSTRUCTS IP DATAGRAM (IP HEADER/TCP HEADER/DATA) TO SEND TO 10.20.20.1"). See also, *id.*, col. 9:8-10.

Finally, the '812 specification describes the construction of the IP Datagram in two places. (See '812 patent, Fig. 8A, col. 9:46-57 and Fig. 8B, col. 10:1-11.)

Thus, the '812 specification discloses structure and the corresponding algorithm for causing a computer to perform the recited function, and the Court should adopt this construction.

PSI again asserts that the '812 specification provides "no corresponding structure" for this element. This is incorrect, as there is ample disclosure in the specification linking the structure cited by IBM to this function. PSI's position is thus contrary to the plain language of the '812 specification and should be rejected.

(iii) "computer readable program code means for causing a computer to effect determining whether said destination IP address for said IP datagram comprises an IP address saved at said host-network interface for said at least one partition, and if so, forwarding the IP datagram directly from said first partition to said second partition of said multiple partitions without employing said network" (claim 11)

The parties agree this phrase is governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Causing a computer to effect determining whether the destination IP address for the IP datagram comprises an IP address saved at the host-network interface for said at least one partition, and if so, forwarding the IP datagram directly from the first partition to the second partition of said multiple partitions without employing the network.</p> <p>Structure: An article of manufacture (<i>see, e.g.</i>, 10:46-57) that causes a computer to effect determining whether the destination IP address for the IP datagram comprises an IP address saved at the host-network interface for said at least one partition, and if so, forwarding the IP datagram directly from the first partition to the second partition of said multiple partitions without employing the network (<i>see, e.g.</i>, Fig. 5, 8:20-30; or steps 210-240 of Fig. 6, 8:37-48), and equivalents thereof.</p>	<p>Function: Causing a computer to effect determining whether the destination IP address for the IP datagram comprises an IP address saved at the host-network interface for said at least one <u>partition</u>, and if so, forwarding the IP datagram directly from the first partition to the second partition of said multiple partitions without employing the network.</p> <p>Structure: No corresponding structure.</p>

Structure. The agreed-upon function is performed by the computer readable program code stored on a computer usable medium. The '812 specification discloses the structure for this computer usable medium:

Additionally, at least one program storage device readable by machine, tangibly embodying at least one program of instructions executable by the machine, to perform the capabilities of the present invention, can be provided.

('812 patent, col. 10:53-57.)

The '812 specification also discloses an algorithm for this computer readable program code to cause a computer to perform the recited function, as follows:

In accordance with a further aspect of the invention, these [HOME] entries are preferably used to route packets from a first logical partition (LPAR) to a second LPAR of the host system without going onto a network. When a packet is received from the Host, the destination IP address is "looked up" in the ARP tables of the host-network interface. If an entry is found and it is marked as a HOME entry, then the IP packet is routed directly to the LPAR owning that address. Since the packet is not sent out onto a network, no media or network header needs to be constructed.

('812 patent, col. 8:20-30.) This algorithm is also set forth in steps 210-240 of Fig. 6. (*See also*, *id.*, col. 8:37-48.) Thus, the '812 specification discloses structure and the corresponding algorithm for causing a computer to perform the recited function, and the Court should adopt this construction.

PSI again asserts that the '812 specification provides "no corresponding structure" for this element. This is incorrect, as there is ample disclosure in the specification linking the structure cited by IBM to this function. PSI's position is thus contrary to the plain language of the '812 specification and should be rejected.

B. The '002 "Firmware Booting" Patent

Partitions of a single computer act as if they are separate computers and act independently of the other partitions. When a computer is turned on or "booted," the individual partitions all rely on the same software or "firmware image" to initiate and perform the boot-up process. Thus, if one partition subsequently needs to be re-booted, the computer must re-boot all the other partitions as well.

The '002 patent recognized this problem and disclosed the use of multiple firmware images (e.g., one per partition) so that one partition can re-boot without having to re-boot all of the other partitions.

The parties dispute the construction of the claim terms "firmware image," "storing a plurality of different firmware images in said computer system," and "capable of being executed during a power-on process to boot said computer system." The parties also dispute the identification of structure in the specification of the '002 patent that corresponds to certain limitations of claim 9 that are in means-plus-function format.

1. "firmware image" (claims 1, 9, 17)

IBM's CONSTRUCTION	PSI's CONSTRUCTION
Software copied from non-volatile memory and used to boot a partition.	An instance of a particular firmware. PSI construes "firmware" as "software' stored in a memory chip that holds its content without electrical power, such as, for example, read-only memory (ROM), programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), and non-volatile random access memory (non-volatile RAM)."

The defining characteristic of a firmware image is what it does, as reflected in IBM's proposed construction, not whether it is stored in a chip, as PSI proposes. IBM's construction of "firmware image" is supported by the '002 claims, specification, and prosecution history. IBM will now address the two disputed components of its proposed construction: (a) copied from non-volatile memory, and (b) used to boot a partition (the parties agree that a "firmware image" is software.)

(a) copied from non-volatile memory

The '002 claims recite that a computer system stores a plurality of firmware images (*see, e.g.*, '002 patent, claim 1, col. 8:35-36), but do not limit the type of memory in which those firmware images are stored. The '002 specification describes a preferred embodiment of the invention (*id.*, col. 2:50-51), and states for this preferred embodiment that the "firmware images are stored in non-volatile memory." *Id.*, col. 2:61-63. The '002 specification also provides an exemplary block diagram of the invention (*id.*, Fig. 3), and states that:

A non-volatile memory device 291, such as a DRAM device, is included within service processor 291. The partition tables and firmware images described herein, as well as other information, are stored within service processor memory 291.

('002 patent, col. 6:12-16.) The '002 specification also indicates that a firmware image is copied from the computer system's memory to a partition's memory before booting (or re-booting) a

partition. *Id.*, col. 3:4-9; 7:11-13; 7:40-41. Thus, according to the '002 intrinsic evidence, a firmware image is software stored in non-volatile memory and copied from that non-volatile memory to boot a partition.

PSI, on the other hand, relies on a unduly narrow construction of "firmware"¹³ to attempt to limit the storage of a "firmware image" to a non-volatile memory *chip*. PSI bases its construction on language from the '002 specification, but takes this language out of context. The language that PSI relies upon relates to "partition management firmware" (e.g., the hypervisor), which is different than a firmware image. The entire section reads as follows:

Partition management firmware (hypervisor) 210 performs a number of functions and services for partitions 201-204 to create and enforce the partitioning of logically partitioned platform 200. *Hypervisor 210* is a firmware implemented virtual machine identical to the underlying hardware. Firmware is "software" stored in a memory chip that holds its content without electrical power, such as, for example, read-only memory (ROM), programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), and non-volatile random access memory (non-volatile RAM). Thus, hypervisor 210 allows the simultaneous execution of independent OS images 201a-204a by virtualizing all the hardware resources of logically partitioned platform 200. Hypervisor 210 may attach I/O devices through I/O adapters 248-262 to single virtual machines in an exclusive mode for use by one of OS images 201a-204a.

('002 patent, col. 6:17-33, emphasis added). While the '002 specification's preferred embodiment states that "partition management firmware (hypervisor)" is stored on a memory chip, it does not mention "firmware images" in this context. As discussed above, the '002 specification only states that "firmware images" are stored in non-volatile memory. *See, id.*, col. 2:61-63; 6:12-16. Even if the partition management firmware definition applied to firmware images (which it does not), the fact that it is only a preferred embodiment indicates that the claims should not be so limited. *See Phillips*, 415 F.3d at 1323 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments. . . . In particular, we have expressly rejected the contention that if a patent

describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.") (citations omitted); *see also Comark Communications, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) ("although the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.") (*quoting Constant v. Adv. Micro-Devices, Inc.*, 848 F. 2d 1560, 1571 (Fed. Cir. 1998)).

PSI asserts this narrow construction because its infringing firmware images are stored on a non-volatile *disk* (as opposed to a non-volatile memory *chip*). As with most of the distinctions PSI is now trying to make, this points stems from the fact that PSI's system is an emulator, and thus the PSI z/Architecture firmware is not stored in the actual chips of the underlying Intel processor. A person of ordinary skill in the art would understand that there is no discernible difference between these types of non-volatile memory for purposes of this patent (Smotherman Decl., ¶¶ 22-23), and the '002 patent explicitly recites that such nominal differences are covered by the '002 patent:

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

('002 patent, col. 8:21-30.)

(b) and used to boot a partition

Finally, the '002 claims, specification, and prosecution history all state that a "firmware image" is used to boot (or re-boot) a partition. ('002 patent, col. 1:60-61.) For example, all of the claims recite using one of a plurality of firmware images to re-boot a partition. This functional aspect of the "firmware image" is reflected in the title of the '002 patent, as well as the

¹³ The term "firmware" is only used as a modifier in the phrase "firmware image" and does not need to be construed separately.

summary of the invention, which states that the invention disclosed is used "for booting a partition using one of multiple, different firmware images." *Id.*, col. 2:7-10. Thus, the key feature of the "firmware image" as described in the '002 patent is that it is used to boot the partition.

For these reasons, IBM's proposed construction is consistent with the '002 patent claims, specification, and prosecution file history, and should be adopted.

- 2. "storing a plurality of different firmware images in said computer system/ a plurality of different firmware images being stored in said computer system" (claims 1, 9 and 17)**

IBM's CONSTRUCTION	PSI's CONSTRUCTION
Storing more than one distinct firmware image in the computer system.	Storing at least two firmware images that are not the same as each other.

The parties' respective constructions are extremely close, and IBM is willing to agree to PSI's proposal.

- 3. "capable of being executed during a power-on process to boot said computer system" (claims 1, 9, 17)**

IBM's CONSTRUCTION	PSI's CONSTRUCTION
Capable of being executed during the operations performed by a computer system from the time it is turned on until it is ready to run applications to boot said computer system.	Able to be executed during the operations performed by a computer system from the time it is turned on until it begins executing the partition firmware.

The dispute relating to this phrase is directed to the meaning of "power-on process." IBM construes the "power-on process" as being from the time a computer system is turned on *until it is ready to run applications*. PSI construes the power-on process as being from the time a computer system is turned on *until it begins executing the partition firmware*. The parties dispute *when* the power-on process is complete.

A plain reading of the disputed limitation in Claim 1 refutes PSI's construction:

storing a plurality of different firmware images in said computer system, *each one of said plurality of different firmware images* capable of being executed during a power-on process to boot said computer system;

('002 patent, col. 8:35-38, emphasis added.) The stated "each one of said plurality of different firmware images" (*i.e.*, each firmware image) is the thing that is "capable of being executed during a power-on process to boot said partition." But under PSI's construction, a firmware image could *never* be executed during a power-on process because PSI proposes that the power-on process is complete before (or at the same time that) the computer system *begins* executing this firmware image.

The '002 specification plainly supports this point. Fig. 4 of the '002 specification illustrates a flow chart for booting a partition during the power-on process. ('002 patent, col. 7:1-7.) Block 402 illustrates the system being powered-on, while the remaining blocks illustrate the other parts of the power-on boot process, such as the determination whether a partition needs re-booting (block 416), and execution of the partition firmware to boot or re-boot partition(s) (block 430). After an execution of partition firmware, the flow chart cycles back to block 420 to determine if any other partitions need to be re-booted. Contrary to PSI's proposed construction, it is clear that the power-on process includes the execution of partition firmware, as the '002 claims require.

Finally, PSI's construction simply reads out the claim language "to boot said computer system," which is the function performed by the firmware image and is included in IBM's construction.

Accordingly, the Court should adopt IBM's proposed construction.

4. "instruction means for storing a plurality of different firmware images in said computer system" (claim 9)

The parties agree this phrase is governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Instructing the computer to store a plurality of different firmware images in the computer system.</p> <p>Structure: A computer program product stored in a computer recordable-type media (<i>see, e.g.,</i> 8:4-20) that instructs a computer to store a plurality of different firmware images in the computer system (<i>see, e.g.,</i> Fig. 3, 6:13-16), and equivalents thereof.</p>	<p>Function: Storing a plurality of different firmware images in said computer system.</p> <p>Structure: No corresponding structure.</p>

Function. The function for this element is set forth in the language of the claim element: "instructing the computer to store a plurality of different firmware images in the computer system." The plain and ordinary meaning of this function is easily understood and requires no further construction beyond the other terms already construed.

PSI's proposed construction is nearly identical to IBM's construction, but improperly reads out the word "instruction" and merely recites the claim language after the "instruction means for" clause. The phrase "instruction means" implies that one or more instructions (stored in a computer readable-type media) are used to perform the specified function. IBM's proposed construction incorporates the "instruction" aspect by including the phrase "instructing the computer to" before the specified function.

Accordingly, the Court should adopt IBM's proposed construction.

Structure. As discussed above, the recited function is performed by the instructions stored in a computer recordable-type media. The '002 specification discloses the structure for this computer recordable-type media:

Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions.

('002 patent, col. 8:12-18.)

The '002 specification also discloses the simple algorithm for instructing the computer to store a plurality of different firmware images in the computer system. The '002 specification states that the "firmware images described herein, as well as other information, are stored within service processor memory 291." ('002 patent, col. 6:13-16; *see also*, Fig. 3.)

Thus, the '002 specification discloses structure and an algorithm for instructing a computer to perform the recited function, and the Court should adopt this construction.

PSI asserts that the '002 specification provides "no corresponding structure." This is contrary to the plain language of the '002 specification and should be rejected.

5. "instruction means for rebooting one of said plurality of partitions utilizing one of said plurality of firmware images without rebooting other ones of said plurality of partitions" (claim 9)

The parties agree this phrase is governed by 35 U.S.C. § 112, ¶ 6 (IBM previously disputed this).

IBM's CONSTRUCTION	PSI's CONSTRUCTION
<p>Function: Instructing a computer to reboot one of the plurality of partitions utilizing one of the plurality of firmware images without rebooting other partitions.</p> <p>Structure: A computer program product stored in a computer recordable-type media (<i>see, e.g.</i>, 8:4-20) that instructs a computer to reboot one of the plurality of partitions utilizing one of the plurality of firmware images without rebooting other partitions (<i>see, e.g.</i>, Fig. 3, 6:58-67; or blocks 422-430 of Fig. 4, 7:30-45), and equivalents thereof.</p>	<p>Function: Rebooting one of said plurality of partitions utilizing one of said plurality of firmware images without rebooting other ones of said plurality of partitions.</p> <p>Structure: No corresponding structure.</p>

Function. The function for this element is set forth in the language of the claim element: "instructing a computer to reboot one of the plurality of partitions utilizing one of the plurality of firmware images without rebooting other partitions." The plain and ordinary meaning of this function is easily understood and requires no further construction beyond the other terms already construed.

PSI's proposed construction is nearly identical to IBM's construction, but improperly reads out the word "instruction" and merely recites the claim language after the "instruction means for" clause. As discussed above, the phrase "instruction means" implies that one or more instructions (stored in a computer readable-type media) are used to perform the specified function. IBM's proposed construction incorporates the "instruction" aspect by including the phrase "instructing the computer to" before the specified function.

Accordingly, the Court should adopt IBM's proposed construction.

Structure. As discussed above, the recited function is performed by the instructions stored in a computer recordable-type media. The '002 specification discloses the structure for this computer recordable-type media:

Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions.

('002 patent, col. 8:12-18.)

The '002 specification also discloses an algorithm for instructing the computer to reboot one of the plurality of partitions utilizing one of the plurality of firmware images without rebooting other partitions. For example, blocks 422-430 of Fig. 4 describe the steps for performing this function. (See '002 patent, col. 7:30-45; 6:58-67; Fig. 3.)

Thus, the '002 specification discloses structure and the corresponding algorithm for instructing a computer to perform the recited function, and the Court should adopt this construction.

PSI again asserts that the '002 specification provides "no corresponding structure" for this element. This is incorrect, as there is ample disclosure in the specification linking the structure cited by IBM to this function. PSI's position is thus contrary to the plain language of the '002 specification and should be rejected.

X. THE '851 I/O PATENT

Computers use I/O devices to manage large amounts of data. The most common example is an external disk drive. Since these input/output devices are slow compared to processors, many computer use an I/O subsystem to handle the I/O-related tasks. An "I/O channel" facilitates communication with either I/O processors or I/O devices. Mainframe computers must be capable of managing large amounts of data. If an emulator is to mimic an IBM mainframe, it must have substantial input/output ("I/O") capabilities.

The '851 patent teaches a method of sharing I/O channels and I/O devices by providing multiple control blocks, each of which presents an image of the I/O device to each operating system. This invention significantly increases the number of sharable I/O channels and devices without requiring an increase in the number of physical channels, devices, or control units connected to the computer.

The invention of the '851 patent provides an efficient way for multiple operating systems to share the same I/O resources, such as an external storage device or an I/O channel to such a device. ('851 patent, col. 6:44-45.) The feature is directed to the implementation of what is known as IBM's Multiple Image Facility. Each shared I/O resource has a sharing set of "input/output control blocks" in which a respective control block is assigned to an operating system. ('851 patent, Abstract.)

1. **"input/output control blocks" (claims 9, 21, 22)**

The sole claim construction dispute between the parties involves the construction of the claim phrase "input/output control blocks."

IBM's CONSTRUCTION	PSI's CONSTRUCTION
Data structures containing information about I/O resources.	A hardware or microprogramming construct which specifies a shared resource to an OS, and may be said to represent an image of the resource to each sharing OS.

IBM's construction is supported by the plain and ordinary meaning of the terms "control block" and "input/output," the specification, and the other agreed-upon constructions between the parties for the '851 patent. PSI's construction improperly attempts to read unnecessary structural limitations into the claim by attempting (1) to limit control blocks to a "hardware or microprogramming construct," and (2) to limit the type of information contained in the control block.

Control Block. A control block is simply a data structure in a storage area that holds some type of control information. In the '851 patent, the "input/output control blocks" are data structures used by the input/output subsystem that contain control information about input/output resources. ('851 patent, col. 8:1-2; 12:16-20.) The '851 patent teaches a novel method for increasing the number of input/output resources available to a program through the use of these data structures. *Id.*, col. 1:54-61; 7:43-60; 14:62-15:3.

While PSI correctly states that a control block is a "construct" synonymous with a "data structure," PSI is improperly reading in a limitation from the specification by requiring the control block to be a "hardware or microprogramming construct." Although in one embodiment of the patent the control blocks are referred to as "hardware or micro-programming constructs" ('851 patent, col. 7:60-64), that does not mean that this language should be read into the claim. *See Phillips*, 415 F.3d at 1323 ("[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments. . . . In particular, we have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.") (citations omitted); *see also, Comark Commc'ns*, 156 F.3d at 1187. Rather than being definitional, this reference to control blocks merely indicates that in an embodiment of the invention, the control blocks (*i.e.*, the data structures) are used by microcode to control hardware resources. Data structures, such as the control blocks shown in Figures 12-14, are programming abstractions and the type of programming or programming language that uses them is immaterial. In other words, the same data structure (here, the I/O control block) can exist

exist equally as well in a microprogram, a z/Architecture program, an Intel® Itanium® program, or any other type of program.

In all, the plain and ordinary meaning of control block is not limited by form – a control block can be operated on by hardware, software, firmware, microcode, or any other structure allowed to access the appropriate storage area.

Input/Output. IBM's construction correctly sets forth that the "input/output control blocks" contain information about input/output resources. Nothing more is required to define the control information associated with these control blocks. The title of the '851 patent is "Method And Means For Sharing *I/O Resources* By a Plurality of Operating Systems." The "I/O resources" are shared by using the input/output control blocks, which not surprisingly contain information about input/output ("I/O") resources. "It is the use of these control blocks by the I/O subsystem and the CU which permits the different OSs to access and directly share all the same *I/O resources*." ('851 patent, col. 14:51-54, emphasis added.)

PSI's construction generally accepts this premise, but is unnecessarily limiting and again improperly imports limitations from the specification into the construction. PSI's construction that the control blocks specify "a shared resource to an OS, and may be said to represent an image of the resource to each sharing OS" adds pointless limitations given the plain and ordinary meaning of the term. While the language relied upon by PSI is taken from the specification ('851 patent, col. 7:43-48), this language is only meaningful in the context of the entire discussion where it appears. There is absolutely no clarity gained by construing the "input/output control block" data structures as "specifying" a shared resource or as ones that "may be said" to represent an image of the resource to each sharing OS. *FieldTurf USA, Inc. v. Sports Const. Group LLC*, 499 F.Supp.2d 907, 926 (refusing to import statements from the specification into the claims because the statements "would not serve to help the jury but would instead cause confusion").

Furthermore, PSI accepts IBM's position that "input/output control blocks" contain information about I/O resources in a separate construction already agreed upon by the parties. The manner by which I/O resources are shared in the '851 patent is by grouping the input/output

control blocks in "sharing sets." As agreed upon by IBM and PSI, a "sharing set" is "a set of control blocks in which each such control block corresponds to *the same* input/output resource." It would logically follow that, generally, a control block corresponds to *an* input/output resource and therefore would "contain information about input/output resources" as set forth by IBM's construction.

In sum, "input/output control blocks" should be construed to mean "data structures containing information about I/O resources" as correctly set forth by IBM.

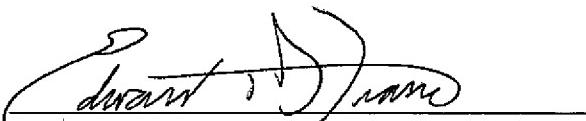
XI. CONCLUSION

For the foregoing reasons, IBM respectfully requests that the Court adopt IBM's proposed constructions for the disputed terms and phrases.

Dated: April 21, 2008

QUINN EMANUEL URQUHART OLIVER
& HEDGES, LLP

By:



Richard I. Werder, Jr.
Edward J. DeFranco
Richard W. Erwine
Alexander Rudis
51 Madison Avenue
22nd Floor
New York, New York 10010-1601
(212) 849-7000

Frederick A. Lorig
865 S. Figueroa Street
Los Angeles, California 90017
(213) 443-3000

Attorneys for Plaintiff
International Business Machines Corporation